selçuk üniversitesi

mühendislik fakültesi

elektrik-elektronik bölümü

robotiğe giriş dersi

proje ödevi raporu

konu:puma 560 incelenmesi

ögrenci:soner tuna

no:141222127

içindekiler

1.özet

2.giriş puma 560

3.link parametlerin çıkarılışı

4.matlap kodları

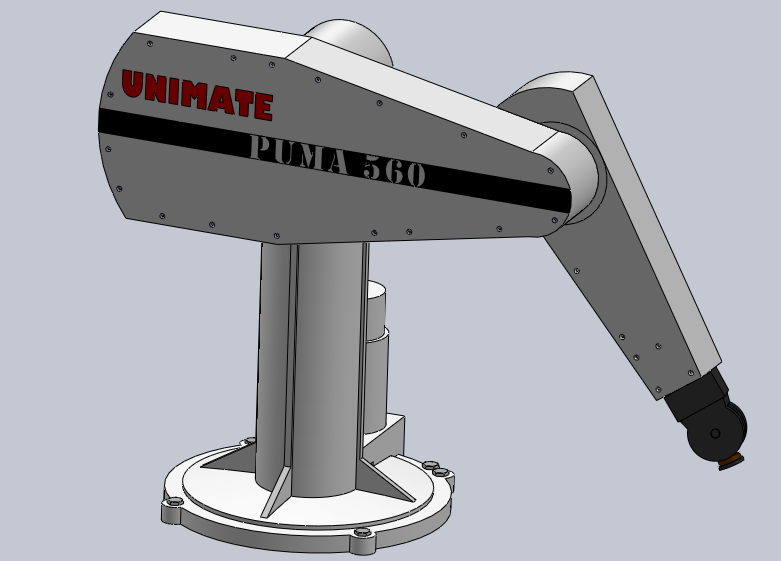
4.kaynaklar

**ÖZET**

Teknolojinin bu kadar gelişmesine ve ilerlemesiyle robotlar günümüzde hızlı bir şekilde gelişmekte ve çoğalmaktadır. Her türlü alanda bir robotlar yer almakta ve çok geniş alanlarda kullanılmaktır. Bu robotların gelişmesiyle birlik de sanayiler de robot eller çoğalmaya başlamıştır. Bu robot ellerin çoğalmaya başlamasıyla sanayide nerdeyse bir devrim olmuş ve sanayi alanda çok kayda değer bir gelişme gözlenmiştir. Çünkü bu insanların yapabileceği işleri daha kısa sürede ve daha hassasiyetli bir şekilde yapılmaya başlanmıştır. Robot eller sayesinde kısa sürede bir çok işi seri bir şekilde yapılmaya başlanmış ve günümüz dede her yere robot eller tasarlanmaya başlanmıştır. Robot eller sayesinde ürünlerde de hassasiyetli bir şekilde tüm işlemleri gerçekleştiği için ürün zahiyatların önüne geçilmiş ve daha kaliteli ürünler piyasaya sürülmüştür. Bu robot eller sayesinde bir ürün artık ham madde olarak bir yerden girip diğer taraf dan ürünün hiçbir el değmeden piyasaya sürürmektedir. İşte robot ellerin tüm bu avantajları için mühendisler bu robot el üzerine bir çok robot tasarlamışlar ve geliştirmişlerdir.Robot ellerin her türlü ihtiyaçlarını karşılıyabilmek için bir çok formüller çıkarmışlar ve daha çok kullanılabilirliğini artırabilmek için bir çok hesaplamalar yapmışlardır. İşte bizim bu yazımımızda da bir robot el olan puma 560 robot eli her türlü yönü ile incelenmiş ve tanıtılmaya çalışılmıştır.

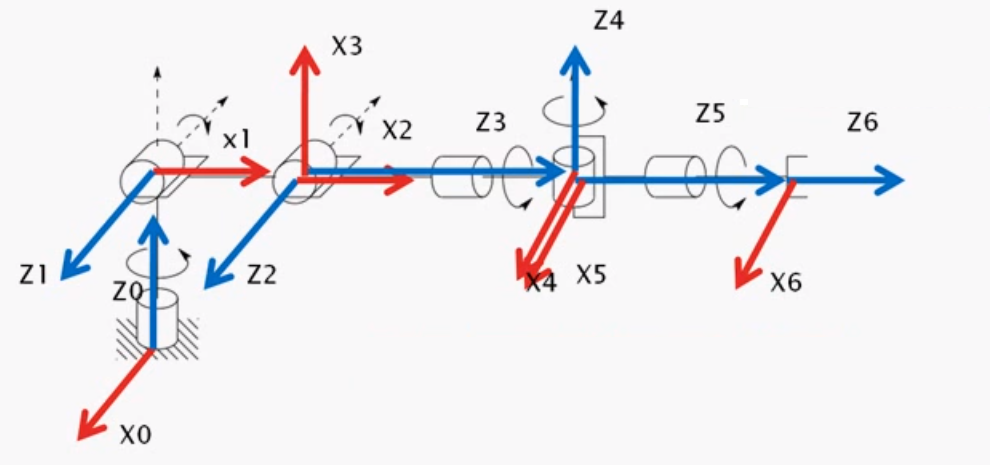
**PUMA 560**

Puma 560 robot şekil 1’de görülmektedir. Şekil 1'de gösterilen PUMA 560, 6 joint den oluşmakdatır ve altı derecelik bir serbest dönme alana sahiptir. Puma 560 robot manipülaörümüz 6 adet servo motorlar yardımıyla hareketi sağlanmaktadır. Robot manipülatörler geometrisine veya kinematik yapısına uygulama tipine ve hangi tip kontrol ile kontrol edildiği gibi birçok kriterlerle gore sınıflandırılır. Bunlar eklemli, küresel, silindirik, kartezyen, gibi kriterlere arılır puma 560 da bu sınıfdan silindirik manipulator içine girer.

şekil 1

**Link Parametrelerin Çıkarılışı**

Şimdi robotumuzun link parametrelerini çıkaralım. Önce robotumuzun anlaşılır bir şekilde şekil 2’de görüldüğü gibi eksenlerini yerleştirelim daha sonra terimlerimizin ne anlama geldiğine bakalım ai ve alpha z eksenine göre işlemler yerleştirilecek, di ve thetai ise x eksenine bakarak yerleştirmemizi yapacağız. Şimdi birkaç parametremizin çıkarımını bakalım.1. parametremizin ai =x0 boyunca z0 ile z1 arasındaki mesafe 0, alpha=x0 etrafında z0 ile z1 arasındaki açı 90 derece,di=z1 boyunca x0 ile x1 arasındaki mesafe l1,th1=z1 boyunca x0 ile x1 arasındaki açı diyere diğer parametrelerimizi de şekil 3’de görüldüğü gibi bu yöntemle daha kolay bir şekilde çıkardık.

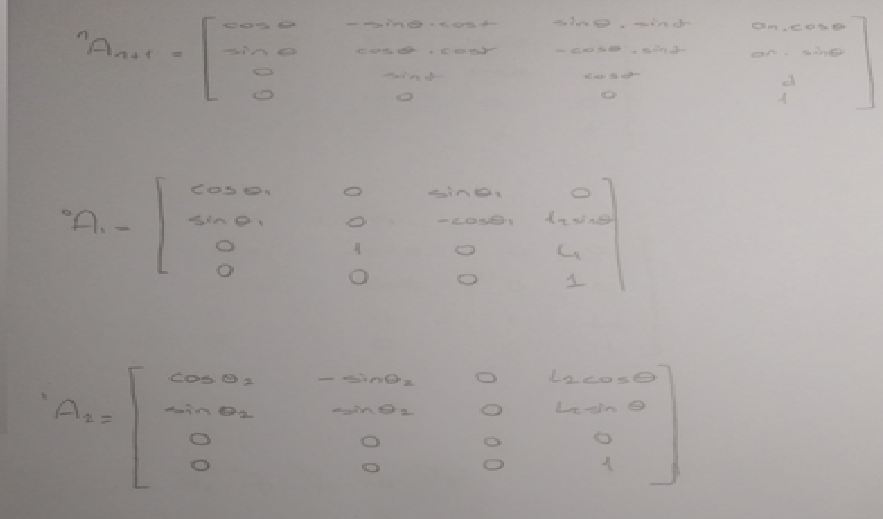


Şekil 2

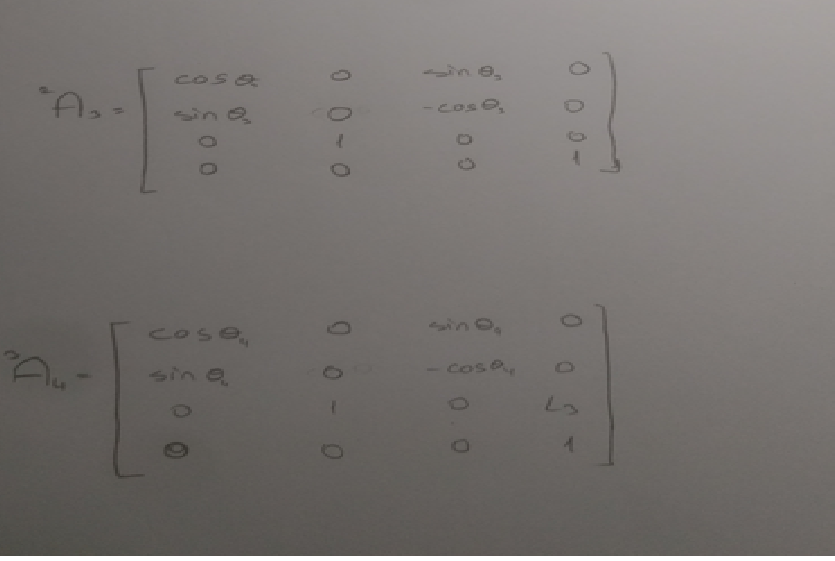


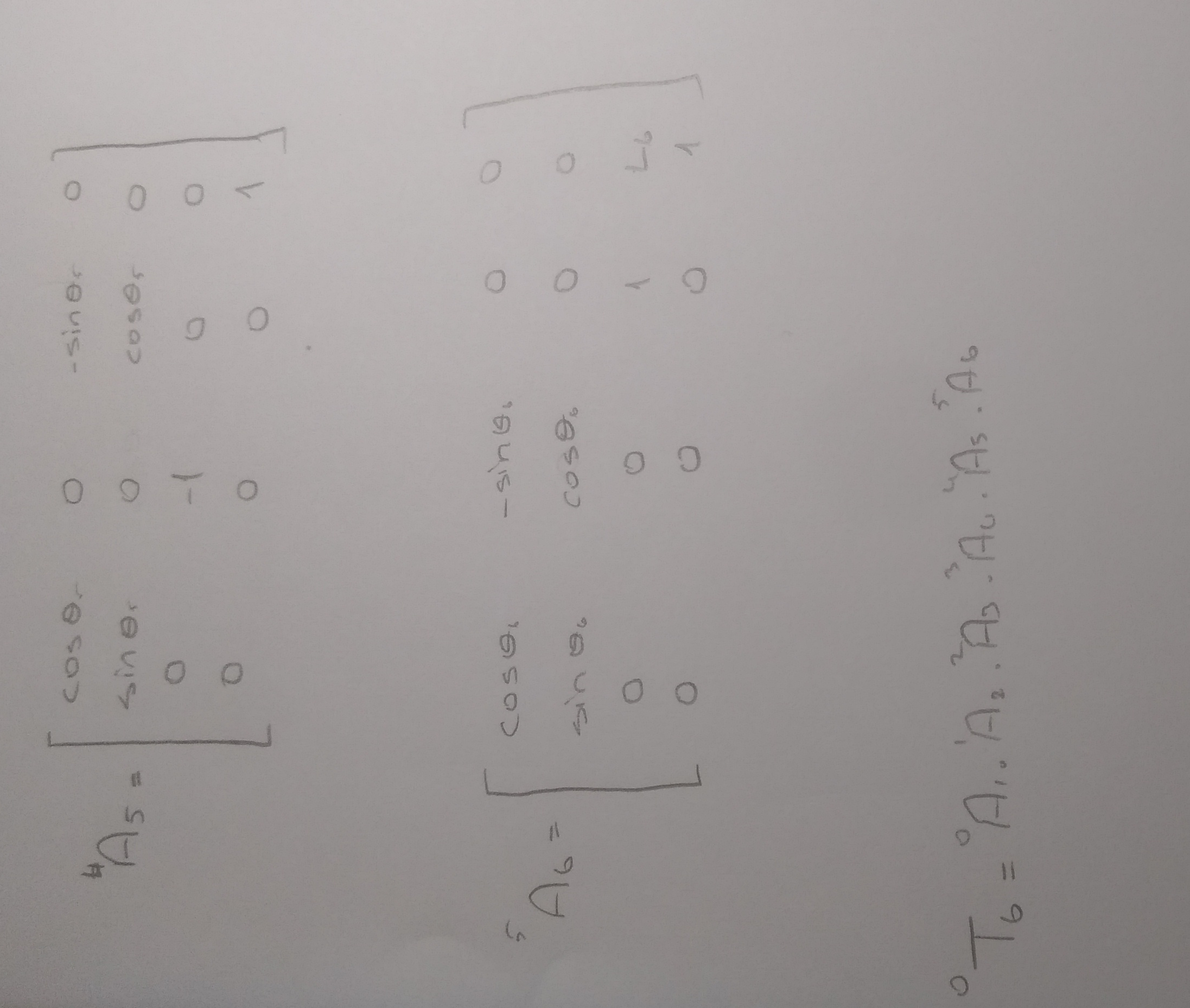
Şekil 3

Şimdi çıkarmış olduğumuz bu link parametrelerin hepsinin tek tek transfer formunu bakalım. Şekil 4,5,6 de görüldüğü gibi her bir linki hesapladık ve transfer formu bu altısının çarpımı sonucu bulunacaktır.



Şekil 4



Şekil 5

Şekil 6

**Puma 560 matlap kodlar**

% This is PUMA3d.M, a 3D Matlab Kinematic model of a Puma robot located

% in the robotics lab of Walla Walla University.

% The file uses CAD data converted to Matlab using cad2matdemo.m, which

% is located on the Mathworks central file exchange.

%

% This file is still being developed, for the latest version check the

% Mathworks central file exchange.

%

% Todo list:

% 1) optimize pumaANI, lots of stuff in loop that needs help.

% 2) move x, y, and z position to end effecter, not link 6 origin.

% 3) Toggle kinematics buttons on/off with inverse kinematics button.

% 4) Make this work with real time inverse kinematics.

% 5) Make the track on and off option better.

% 6) add other things that makes this program fun.

% 7) check for noplots and nogos

% 8) add some better "demos" for the button

% 9) Fix problem of more than one robot window.

%

function puma3d

% GUI kinematic demo for the Puma Robot.

% Robot geometry uses the CAD2MATDEMO code in the Mathworks file exchange

%

%%

loaddata

InitHome

%

% Create the push buttons: pos is: [left bottom width height]

demo = uicontrol(fig\_1,'String','Demo','callback',@demo\_button\_press,...

'Position',[20 5 60 20]);

rnd\_demo = uicontrol(fig\_1,'String','Random Move','callback',@rnd\_demo\_button\_press,...

'Position',[100 5 80 20]);

clr\_trail = uicontrol(fig\_1,'String','Clr

Trail','callback',@clr\_trail\_button\_press,...

'Position',[200 5 60 20]);

%

home = uicontrol(fig\_1,'String','Home','callback',@home\_button\_press,...

'Position',[280 5 70 20]);

%

% Kinematics Panel

%

K\_p = uipanel(fig\_1,...

'units','pixels',...

'Position',[20 45 265 200],...

'Title','Kinematics','FontSize',11);

%

% Angle Range Default Name

% Theta 1: 320 (-160 to 160) 90 Waist Joint

% Theta 2: 220 (-110 to 110) -90 Shoulder Joint

% Theta 3: 270 (-135 to 135) -90 Elbow Joint

% Theta 4: 532 (-266 to 266) 0 Wrist Roll

% Theta 5: 200 (-100 to 100) 0 Wrist Bend

% Theta 6: 532 (-266 to 266) 0 Wrist Swivel

t1\_home = 90; % offset to define the "home" position as UP.

t2\_home = -90;

t3\_home = -90;

LD = 105; % Left, used to set the GUI.

HT = 18; % Height

BT = 156; % Bottom

%% GUI buttons for Theta 1. pos is: [left bottom width height]

t1\_slider = uicontrol(K\_p,'style','slider',...

'Max',160,'Min',-160,'Value',0,...

'SliderStep',[0.05 0.2],...

'callback',@t1\_slider\_button\_press,...

'Position',[LD BT 120 HT]);

t1\_min = uicontrol(K\_p,'style','text',...

'String','-160',...

'Position',[LD-30 BT+1 25 HT-4]); % L, from bottom, W, H

t1\_max = uicontrol(K\_p,'style','text',...

'String','+160',...

'Position',[LD+125 BT+1 25 HT-4]); % L, B, W, H

t1\_text = uibutton(K\_p,'style','text',... % Nice program Doug. Need this

'String','\theta\_1',... % due to no TeX in uicontrols.

'Position',[LD-100 BT 20 HT]); % L, B, W, H

% t1\_text = uicontrol(K\_p,'style','text',... % when matlab fixes uicontrol

% 'String','t1',... % for TeX, then I can use this.

% 'Position',[LD-100 BT 20 HT]); % L, B, W, H

t1\_edit = uicontrol(K\_p,'style','edit',...

'String',0,...

'callback',@t1\_edit\_button\_press,...

'Position',[LD-75 BT 30 HT]); % L, B, W, H

%

%% GUI buttons for Theta 2.

BT = 126; % Bottom

t2\_slider = uicontrol(K\_p,'style','slider',...

'Max',115,'Min',-115,'Value',0,... % Mech. stop limits !

'SliderStep',[0.05 0.2],...

'callback',@t2\_slider\_button\_press,...

'Position',[LD BT 120 HT]);

t2\_min = uicontrol(K\_p,'style','text',...

'String','-110',...

'Position',[LD-30 BT+1 25 HT-4]); % L, from bottom, W, H

t2\_max = uicontrol(K\_p,'style','text',...

'String','+110',...

'Position',[LD+125 BT+1 25 HT-4]); % L, B, W, H

t2\_text = uibutton(K\_p,'style','text',...

'String','\theta\_2',...

'Position',[LD-100 BT 20 HT]); % L, B, W, H

t2\_edit = uicontrol(K\_p,'style','edit',...

'String',0,...

'callback',@t2\_edit\_button\_press,...

'Position',[LD-75 BT 30 HT]); % L, B, W, H

%

%% GUI buttons for Theta 3.

BT = 96; % Bottom

t3\_slider = uicontrol(K\_p,'style','slider',...

'Max',135,'Min',-135,'Value',0,...

'SliderStep',[0.05 0.2],...

'callback',@t3\_slider\_button\_press,...

'Position',[LD BT 120 HT]);

t3\_min = uicontrol(K\_p,'style','text',...

'String','-135',...

'Position',[LD-30 BT+1 25 HT-4]); % L, from bottom, W, H

t3\_max = uicontrol(K\_p,'style','text',...

'String','+135',...

'Position',[LD+125 BT+1 25 HT-4]); % L, B, W, H

t3\_text = uibutton(K\_p,'style','text',...

'String','\theta\_3',...

'Position',[LD-100 BT 20 HT]); % L, B, W, H

t3\_edit = uicontrol(K\_p,'style','edit',...

'String',0,...

'callback',@t3\_edit\_button\_press,...

'Position',[LD-75 BT 30 HT]); % L, B, W, H

%

%% GUI buttons for Theta 4.

BT = 66; % Bottom

t4\_slider = uicontrol(K\_p,'style','slider',...

'Max',266,'Min',-266,'Value',0,...

'SliderStep',[0.05 0.2],...

'callback',@t4\_slider\_button\_press,...

'Position',[LD BT 120 HT]);

t4\_min = uicontrol(K\_p,'style','text',...

'String','-266',...

'Position',[LD-30 BT+1 25 HT-4]); % L, from bottom, W, H

t4\_max = uicontrol(K\_p,'style','text',...

'String','+266',...

'Position',[LD+125 BT+1 25 HT-4]); % L, B, W, H

t4\_text = uibutton(K\_p,'style','text',...

'String','\theta\_4',...

'Position',[LD-100 BT 20 HT]); % L, B, W, H

t4\_edit = uicontrol(K\_p,'style','edit',...

'String',0,...

'callback',@t4\_edit\_button\_press,...

'Position',[LD-75 BT 30 HT]); % L, B, W, H

%

%% GUI buttons for Theta 5.

BT = 36; % Bottom

t5\_slider = uicontrol(K\_p,'style','slider',...

'Max',100,'Min',-100,'Value',0,...

'SliderStep',[0.05 0.2],...

'callback',@t5\_slider\_button\_press,...

'Position',[LD BT 120 HT]);

t5\_min = uicontrol(K\_p,'style','text',...

'String','-100',...

'Position',[LD-30 BT+1 25 HT-4]); % L, from bottom, W, H

t5\_max = uicontrol(K\_p,'style','text',...

'String','+100',...

'Position',[LD+125 BT+1 25 HT-4]); % L, B, W, H

t5\_text = uibutton(K\_p,'style','text',...

'String','\theta\_5',...

'Position',[LD-100 BT 20 HT]); % L, B, W, H

t5\_edit = uicontrol(K\_p,'style','edit',...

'String',0,...

'callback',@t5\_edit\_button\_press,...

'Position',[LD-75 BT 30 HT]); % L, B, W, H

%

%% GUI buttons for Theta 6.

BT = 6; % Bottom

t6\_slider = uicontrol(K\_p,'style','slider',...

'Max',266,'Min',-266,'Value',0,...

'SliderStep',[0.05 0.2],...

'callback',@t6\_slider\_button\_press,...

'Position',[LD BT 120 HT]);

t6\_min = uicontrol(K\_p,'style','text',...

'String','-266',...

'Position',[LD-30 BT+1 25 HT-4]); % L, from bottom, W, H

t6\_max = uicontrol(K\_p,'style','text',...

'String','+266',...

'Position',[LD+125 BT+1 25 HT-4]); % L, B, W, H

t6\_text = uibutton(K\_p,'style','text',...

'String','\theta\_6',...

'Position',[LD-100 BT 20 HT]); % L, B, W, H

t6\_edit = uicontrol(K\_p,'style','edit',...

'String',0,...

'callback',@t6\_edit\_button\_press,...

'Position',[LD-75 BT 30 HT]); % L, B, W, H

%

%% Slider for Theta 1 motion.

%

function t1\_slider\_button\_press(h,dummy)

slider\_value = round(get(h,'Value'));

set(t1\_edit,'string',slider\_value);

T\_Old = getappdata(0,'ThetaOld');

t2old = T\_Old(2); t3old = T\_Old(3); t4old = T\_Old(4);

t5old = T\_Old(5); t6old = T\_Old(6);

pumaANI(slider\_value+t1\_home,t2old,t3old,t4old,t5old,t6old,10,'n')

end

%

%% Slider for Theta 2 motion.

%

function t2\_slider\_button\_press(h,dummy)

slider\_value = round(get(h,'Value'));

set(t2\_edit,'string',slider\_value);

T\_Old = getappdata(0,'ThetaOld');

t1old = T\_Old(1); t3old = T\_Old(3); t4old = T\_Old(4);

t5old = T\_Old(5); t6old = T\_Old(6);

pumaANI(t1old,slider\_value+t2\_home,t3old,t4old,t5old,t6old,10,'n')

end

%

%% Slider for Theta 3 motion.

function t3\_slider\_button\_press(h,dummy)

slider\_value = round(get(h,'Value'));

set(t3\_edit,'string',slider\_value);

T\_Old = getappdata(0,'ThetaOld');

t1old = T\_Old(1); t2old = T\_Old(2); t4old = T\_Old(4);

t5old = T\_Old(5); t6old = T\_Old(6);

pumaANI(t1old,t2old,slider\_value+t3\_home,t4old,t5old,t6old,10,'n')

end

%

%% Slider for Theta 4 motion.

function t4\_slider\_button\_press(h,dummy)

slider\_value = round(get(h,'Value'));

set(t4\_edit,'string',slider\_value);

T\_Old = getappdata(0,'ThetaOld');

t1old = T\_Old(1); t2old = T\_Old(2); t3old = T\_Old(3);

t5old = T\_Old(5); t6old = T\_Old(6);

pumaANI(t1old,t2old,t3old,slider\_value,t5old,t6old,10,'n')

end

%

%% Slider for Theta 5 motion.

function t5\_slider\_button\_press(h,dummy)

slider\_value = round(get(h,'Value'));

set(t5\_edit,'string',slider\_value);

T\_Old = getappdata(0,'ThetaOld');

t1old = T\_Old(1); t2old = T\_Old(2); t3old = T\_Old(3);

t4old = T\_Old(4); t6old = T\_Old(6);

pumaANI(t1old,t2old,t3old,t4old,slider\_value,t6old,10,'n')

end

%

%% Slider for Theta 6 motion.

function t6\_slider\_button\_press(h,dummy)

slider\_value = round(get(h,'Value'));

set(t6\_edit,'string',slider\_value);

T\_Old = getappdata(0,'ThetaOld');

t1old = T\_Old(1); t2old = T\_Old(2); t3old = T\_Old(3);

t4old = T\_Old(4); t5old = T\_Old(5);

pumaANI(t1old,t2old,t3old,t4old,t5old,slider\_value,10,'n')

end

%

%% Edit box for Theta 1 motion.

%

function t1\_edit\_button\_press(h,dummy)

user\_entry = check\_edit(h,-160,160,0,t1\_edit);

set(t1\_slider,'Value',user\_entry); % slider = text box.

T\_Old = getappdata(0,'ThetaOld'); % Current pose

%

t2old = T\_Old(2); t3old = T\_Old(3); t4old = T\_Old(4);

t5old = T\_Old(5); t6old = T\_Old(6);

%

pumaANI(user\_entry+t1\_home,t2old,t3old,t4old,t5old,t6old,10,'n')

end

%

%% Edit box for Theta 2 motion.

%

function t2\_edit\_button\_press(h,dummy)

user\_entry = check\_edit(h,-110,110,0,t2\_edit);

set(t2\_slider,'Value',user\_entry); % slider = text box.

T\_Old = getappdata(0,'ThetaOld'); % Current pose

%

t1old = T\_Old(1); t3old = T\_Old(3); t4old = T\_Old(4);

t5old = T\_Old(5); t6old = T\_Old(6);

%

pumaANI(t1old,user\_entry+t2\_home,t3old,t4old,t5old,t6old,10,'n')

end

%% Edit box for Theta 3 motion.

%

function t3\_edit\_button\_press(h,dummy)

user\_entry = check\_edit(h,-135,135,0,t3\_edit);

set(t3\_slider,'Value',user\_entry); % slider = text box.

T\_Old = getappdata(0,'ThetaOld'); % Current pose

%

t1old = T\_Old(1); t2old = T\_Old(2); t4old = T\_Old(4);

t5old = T\_Old(5); t6old = T\_Old(6);

%

pumaANI(t1old,t2old,user\_entry+t3\_home,t4old,t5old,t6old,10,'n')

end

%%

%% Edit box for Theta 4 motion.

%

function t4\_edit\_button\_press(h,dummy)

user\_entry = check\_edit(h,-266,266,0,t4\_edit);

set(t4\_slider,'Value',user\_entry); % slider = text box.

T\_Old = getappdata(0,'ThetaOld'); % Current pose

%

t1old = T\_Old(1); t2old = T\_Old(2); t3old = T\_Old(3);

t5old = T\_Old(5); t6old = T\_Old(6);

%

pumaANI(t1old,t2old,t3old,user\_entry,t5old,t6old,10,'n')

end

%% Edit box for Theta 5 motion.

%

function t5\_edit\_button\_press(h,dummy)

user\_entry = check\_edit(h,-100,100,0,t5\_edit);

set(t5\_slider,'Value',user\_entry); % slider = text box.

T\_Old = getappdata(0,'ThetaOld'); % Current pose

%

t1old = T\_Old(1); t2old = T\_Old(2); t3old = T\_Old(3);

t4old = T\_Old(4); t6old = T\_Old(6);

%

pumaANI(t1old,t2old,t3old,t4old,user\_entry,t6old,10,'n')

end

%%

%% Edit box for Theta 6 motion.

%

function t6\_edit\_button\_press(h,dummy)

user\_entry = check\_edit(h,-266,266,0,t6\_edit);

set(t6\_slider,'Value',user\_entry); % slider = text box.

T\_Old = getappdata(0,'ThetaOld'); % Current pose

%

t1old = T\_Old(1); t2old = T\_Old(2); t3old = T\_Old(3);

t4old = T\_Old(4); t5old = T\_Old(5);

%

pumaANI(t1old,t2old,t3old,t4old,t5old,user\_entry,10,'n')

end

%%

function user\_entry = check\_edit(h,min\_v,max\_v,default,h\_edit)

% This function will check the value typed in the text input box

% against min and max values, and correct errors.

%

% h: handle of gui

% min\_v min value to check

% max\_v max value to check

% default is the default value if user enters non number

% h\_edit is the edit value to update.

%

user\_entry = str2double(get(h,'string'));

if isnan(user\_entry)

errordlg(['You must enter a numeric value, defaulting to ',num2str(default),'.'],'Bad Input','modal')

set(h\_edit,'string',default);

user\_entry = default;

end

%

if user\_entry < min\_v

errordlg(['Minimum limit is ',num2str(min\_v),' degrees, using ',num2str(min\_v),'.'],'Bad Input','modal')

user\_entry = min\_v;

set(h\_edit,'string',user\_entry);

end

if user\_entry > max\_v

errordlg(['Maximum limit is ',num2str(max\_v),' degrees, using ',num2str(max\_v),'.'],'Bad Input','modal')

user\_entry = max\_v;

set(h\_edit,'string',user\_entry);

end

end

%

%% Demo button's callback

function demo\_button\_press(h,dummy)

%

% disp('pushed demo bottom');

% R = 500;

% x = 1000;

n = 2; % demo ani steps

num = 30; % home to start, and end to home ani steps

% j = 1;

% M = 1000;

for t = 0:.1:7\*pi

Px = 30\*t\*cos(t);

Py = 1200-300\*t\*(t)/(50\*pi);

Pz = 30\*t\*sin(t);

[theta1,theta2,theta3,theta4,theta5,theta6] = PumaIK(Px,Py,Pz);

if t==0 %move to start of demo

pumaANI(theta1,theta2,theta3-180,0,0,0,num,'n')

end

% Theta 4, 5 & 6 are zero due to plotting at wrist origen.

pumaANI(theta1,theta2,theta3-180,0,0,0,n,'y')

set(t1\_edit,'string',round(theta1)); % Update slider and text.

set(t1\_slider,'Value',round(theta1));

set(t2\_edit,'string',round(theta2));

set(t2\_slider,'Value',round(theta2));

set(t3\_edit,'string',round(theta3-180));

set(t3\_slider,'Value',round(theta3-180));

end

gohome

% pumaANI(90,-90,-90,0,0,0,num,'n')

end

%

%

%%

function home\_button\_press(h,dummy)

%disp('pushed home bottom');

gohome

end

%

%%

function clr\_trail\_button\_press(h,dummy)

%disp('pushed clear trail bottom');

handles = getappdata(0,'patch\_h'); %

Tr = handles(9);

%

setappdata(0,'xtrail',0); % used for trail tracking.

setappdata(0,'ytrail',0); % used for trail tracking.

setappdata(0,'ztrail',0); % used for trail tracking.

%

set(Tr,'xdata',0,'ydata',0,'zdata',0);

end

%

%

function rnd\_demo\_button\_press(h, dummy)

%disp('pushed random demo bottom');

% a = 10; b = 50; x = a + (b-a) \* rand(5)

% Angle Range Default Name

% Theta 1: 320 (-160 to 160) 90 Waist Joint

% Theta 2: 220 (-110 to 110) -90 Shoulder Joint

% Theta 3: 270 (-135 to 135) -90 Elbow Joint

% Theta 4: 532 (-266 to 266) 0 Wrist Roll

% Theta 5: 200 (-100 to 100) 0 Wrist Bend

% Theta 6: 532 (-266 to 266) 0 Wrist Swival

t1\_home = 90; % offsets to define the "home" postition as UP.

t2\_home = -90;

t3\_home = -90;

theta1 = -160 + 320\*rand(1); % offset for home

theta2 = -110 + 220\*rand(1); % in the UP pos.

theta3 = -135 + 270\*rand(1);

theta4 = -266 + 532\*rand(1);

theta5 = -100 + 200\*rand(1);

theta6 = -266 + 532\*rand(1);

n = 50;

pumaANI(theta1+t1\_home,theta2+t2\_home,theta3+t3\_home,theta4,theta5,theta6,n,'y')

set(t1\_edit,'string',round(theta1)); % Update slider and text.

set(t1\_slider,'Value',round(theta1));

set(t2\_edit,'string',round(theta2));

set(t2\_slider,'Value',round(theta2));

set(t3\_edit,'string',round(theta3));

set(t3\_slider,'Value',round(theta3));

set(t4\_edit,'string',round(theta4));

set(t4\_slider,'Value',round(theta4));

set(t5\_edit,'string',round(theta5));

set(t5\_slider,'Value',round(theta5));

set(t6\_edit,'string',round(theta6));

set(t6\_slider,'Value',round(theta6));

end

%%

%Here are the functions used for this robot example:

%

%%

% When called this function will simply initialize a plot of the Puma 762

% robot by plotting it in it's home orientation and setting the current

% angles accordingly.

function gohome()

pumaANI(90,-90,-90,0,0,0,20,'n') % show it animate home

%PumaPOS(90,-90,-90,0,0,0) %drive it home, no animate.

set(t1\_edit,'string',0);

set(t1\_slider,'Value',0); %At the home position, so all

set(t2\_edit,'string',0); %sliders and input boxes = 0.

set(t2\_slider,'Value',0);

set(t3\_edit,'string',0);

set(t3\_slider,'Value',0);

set(t4\_edit,'string',0);

set(t4\_slider,'Value',0);

set(t5\_edit,'string',0);

set(t5\_slider,'Value',0);

set(t6\_edit,'string',0);

set(t6\_slider,'Value',0);

setappdata(0,'ThetaOld',[90,-90,-90,0,0,0]);

end

%%

% This function will load the 3D CAD data.

%

function loaddata

% Loads all the link data from file linksdata.mat.

% This data comes from a Pro/E 3D CAD model and was made with cad2matdemo.m

% from the file exchange. All link data manually stored in linksdata.mat

[linkdata]=load('linksdata.mat','s1','s2', 's3','s4','s5','s6','s7','A1');

%Place the robot link 'data' in a storage area

setappdata(0,'Link1\_data',linkdata.s1);

setappdata(0,'Link2\_data',linkdata.s2);

setappdata(0,'Link3\_data',linkdata.s3);

setappdata(0,'Link4\_data',linkdata.s4);

setappdata(0,'Link5\_data',linkdata.s5);

setappdata(0,'Link6\_data',linkdata.s6);

setappdata(0,'Link7\_data',linkdata.s7);

setappdata(0,'Area\_data',linkdata.A1);

end

%

%%

% Use forward kinematics to place the robot in a specified configuration.

%

function PumaPOS(theta1,theta2,theta3,theta4,theta5,theta6)

s1 = getappdata(0,'Link1\_data');

s2 = getappdata(0,'Link2\_data');

s3 = getappdata(0,'Link3\_data');

s4 = getappdata(0,'Link4\_data');

s5 = getappdata(0,'Link5\_data');

s6 = getappdata(0,'Link6\_data');

s7 = getappdata(0,'Link7\_data');

A1 = getappdata(0,'Area\_data');

%

a2 = 650;

a3 = 0;

d3 = 190;

d4 = 600;

Px = 5000;

Py = 5000;

Pz = 5000;

t1 = theta1;

t2 = theta2;

t3 = theta3 %-180;

t4 = theta4;

t5 = theta5;

t6 = theta6;

%

% Forward Kinematics

T\_01 = tmat(0, 0, 0, t1);

T\_12 = tmat(-90, 0, 0, t2);

T\_23 = tmat(0, a2, d3, t3);

T\_34 = tmat(-90, a3, d4, t4);

T\_45 = tmat(90, 0, 0, t5);

T\_56 = tmat(-90, 0, 0, t6);

%T\_01 = T\_01;

T\_02 = T\_01\*T\_12;

T\_03 = T\_02\*T\_23;

T\_04 = T\_03\*T\_34;

T\_05 = T\_04\*T\_45;

T\_06 = T\_05\*T\_56;

%

Link1 = s1.V1;

Link2 = (T\_01\*s2.V2')';

Link3 = (T\_02\*s3.V3')';

Link4 = (T\_03\*s4.V4')';

Link5 = (T\_04\*s5.V5')';

Link6 = (T\_05\*s6.V6')';

Link7 = (T\_06\*s7.V7')';

handles = getappdata(0,'patch\_h'); %

L1 = handles(1);

L2 = handles(2);

L3 = handles(3);

L4 = handles(4);

L5 = handles(5);

L6 = handles(6);

L7 = handles(7);

%

set(L1,'vertices',Link1(:,1:3),'facec', [0.717,0.116,0.123]);

set(L1, 'EdgeColor','none');

set(L2,'vertices',Link2(:,1:3),'facec', [0.216,1,.583]);

set(L2, 'EdgeColor','none');

set(L3,'vertices',Link3(:,1:3),'facec', [0.306,0.733,1]);

set(L3, 'EdgeColor','none');

set(L4,'vertices',Link4(:,1:3),'facec', [1,0.542,0.493]);

set(L4, 'EdgeColor','none');

set(L5,'vertices',Link5(:,1:3),'facec', [0.216,1,.583]);

set(L5, 'EdgeColor','none');

set(L6,'vertices',Link6(:,1:3),'facec', [1,1,0.255]);

set(L6, 'EdgeColor','none');

set(L7,'vertices',Link7(:,1:3),'facec', [0.306,0.733,1]);

set(L7, 'EdgeColor','none');

end

%%

% This function computes the Inverse Kinematics for the Puma 762 robot

% given X,Y,Z coordinates for a point in the workspace. Note: The IK are

% computed for the origin of Coordinate systems 4,5 & 6.

function [theta1,theta2,theta3,theta4,theta5,theta6] = PumaIK(Px,Py,Pz)

theta4 = 0;

theta5 = 0;

theta6 = 0;

sign1 = 1;

sign3 = 1;

nogo = 0;

noplot = 0;

% Because the sqrt term in theta1 & theta3 can be + or - we run through

% all possible combinations (i = 4) and take the first combination that

% satisfies the joint angle constraints.

while nogo == 0;

for i = 1:1:4

if i == 1

sign1 = 1;

sign3 = 1;

elseif i == 2

sign1 = 1;

sign3 = -1;

elseif i == 3

sign1 = -1;

sign3 = 1;

else

sign1 = -1;

sign3 = -1;

end

a2 = 650;

a3 = 0;

d3 = 190;

d4 = 600;

rho = sqrt(Px^2+Py^2);

phi = atan2(Py,Px);

K = (Px^2+Py^2+Pz^2-a2^2-a3^2-d3^2-d4^2)/(2\*a2);

c4 = cos(theta4);

s4 = sin(theta4);

c5 = cos(theta5);

s5 = sin(theta5);

c6 = cos(theta6);

s6 = sin(theta6);

theta1 = (atan2(Py,Px)-atan2(d3,sign1\*sqrt(Px^2+Py^2-d3^2)));

c1 = cos(theta1);

s1 = sin(theta1);

theta3 = (atan2(a3,d4)-atan2(K,sign3\*sqrt(a3^2+d4^2-K^2)));

c3 = cos(theta3);

s3 = sin(theta3);

t23 = atan2((-a3-a2\*c3)\*Pz-(c1\*Px+s1\*Py)\*(d4-a2\*s3),(a2\*s3-d4)\*Pz+(a3+a2\*c3)\*(c1\*Px+s1\*Py));

theta2 = (t23 - theta3);

c2 = cos(theta2);

s2 = sin(theta2);

s23 = ((-a3-a2\*c3)\*Pz+(c1\*Px+s1\*Py)\*(a2\*s3-d4))/(Pz^2+(c1\*Px+s1\*Py)^2);

c23 = ((a2\*s3-d4)\*Pz+(a3+a2\*c3)\*(c1\*Px+s1\*Py))/(Pz^2+(c1\*Px+s1\*Py)^2);

r13 = -c1\*(c23\*c4\*s5+s23\*c5)-s1\*s4\*s5;

r23 = -s1\*(c23\*c4\*s5+s23\*c5)+c1\*s4\*s5;

r33 = s23\*c4\*s5 - c23\*c5;

theta4 = atan2(-r13\*s1+r23\*c1,-r13\*c1\*c23-r23\*s1\*c23+r33\*s23);

r11 = c1\*(c23\*(c4\*c5\*c6-s4\*s6)-s23\*s5\*c6)+s1\*(s4\*c5\*c6+c4\*s6);

r21 = s1\*(c23\*(c4\*c5\*c6-s4\*s6)-s23\*s5\*c6)-c1\*(s4\*c5\*c6+c4\*s6);

r31 = -s23\*(c4\*c5\*c6-s4\*s6)-c23\*s5\*c6;

s5 = -(r13\*(c1\*c23\*c4+s1\*s4)+r23\*(s1\*c23\*c4-c1\*s4)-r33\*(s23\*c4));

c5 = r13\*(-c1\*s23)+r23\*(-s1\*s23)+r33\*(-c23);

theta5 = atan2(s5,c5);

s6 = -r11\*(c1\*c23\*s4-s1\*c4)-r21\*(s1\*c23\*s4+c1\*c4)+r31\*(s23\*s4);

c6 = r11\*((c1\*c23\*c4+s1\*s4)\*c5-c1\*s23\*s5)+r21\*((s1\*c23\*c4-c1\*s4)\*c5-s1\*s23\*s5)-r31\*(s23\*c4\*c5+c23\*s5);

theta6 = atan2(s6,c6);

theta1 = theta1\*180/pi;

theta2 = theta2\*180/pi;

theta3 = theta3\*180/pi;

theta4 = theta4\*180/pi;

theta5 = theta5\*180/pi;

theta6 = theta6\*180/pi;

if theta2>=160 && theta2<=180

theta2 = -theta2;

end

if theta1<=160 && theta1>=-160 && (theta2<=20 && theta2>=-200) && theta3<=45 && theta3>=-225 && theta4<=266 && theta4>=-266 && theta5<=100 && theta5>=-100 && theta6<=266 && theta6>=-266

nogo = 1;

theta3 = theta3+180;

break

end

if i == 4 && nogo == 0

h = errordlg('Point unreachable due to joint angle constraints.','JOINT ERROR');

waitfor(h);

nogo = 1;

noplot = 1;

break

end

end

end

end

%

%%

function pumaANI(theta1,theta2,theta3,theta4,theta5,theta6,n,trail)

% This function will animate the Puma 762 robot given joint angles.

% n is number of steps for the animation

% trail is 'y' or 'n' (n = anything else) for leaving a trail.

%

%disp('in animate');

a2 = 650; %D-H paramaters

a3 = 0;

d3 = 190;

d4 = 600;

% Err2 = 0;

%

ThetaOld = getappdata(0,'ThetaOld');

%

theta1old = ThetaOld(1);

theta2old = ThetaOld(2);

theta3old = ThetaOld(3);

theta4old = ThetaOld(4);

theta5old = ThetaOld(5);

theta6old = ThetaOld(6);

%

t1 = linspace(theta1old,theta1,n);

t2 = linspace(theta2old,theta2,n);

t3 = linspace(theta3old,theta3,n);% -180;

t4 = linspace(theta4old,theta4,n);

t5 = linspace(theta5old,theta5,n);

t6 = linspace(theta6old,theta6,n);

n = length(t1);

for i = 2:1:n

% Forward Kinematics

%

T\_01 = tmat(0, 0, 0, t1(i));

T\_12 = tmat(-90, 0, 0, t2(i));

T\_23 = tmat(0, a2, d3, t3(i));

T\_34 = tmat(-90, a3, d4, t4(i));

T\_45 = tmat(90, 0, 0, t5(i));

T\_56 = tmat(-90, 0, 0, t6(i));

%

% % T\_67 = [ 1 0 0 0

% % 0 1 0 0

% % 0 0 1 188

% % 0 0 0 1];

%T\_01 = T\_01; % it is, but don't need to say so.

T\_02 = T\_01\*T\_12;

T\_03 = T\_02\*T\_23;

T\_04 = T\_03\*T\_34;

T\_05 = T\_04\*T\_45;

T\_06 = T\_05\*T\_56;

% T\_07 = T\_06\*T\_67;

%

s1 = getappdata(0,'Link1\_data');

s2 = getappdata(0,'Link2\_data');

s3 = getappdata(0,'Link3\_data');

s4 = getappdata(0,'Link4\_data');

s5 = getappdata(0,'Link5\_data');

s6 = getappdata(0,'Link6\_data');

s7 = getappdata(0,'Link7\_data');

%A1 = getappdata(0,'Area\_data');

Link1 = s1.V1;

Link2 = (T\_01\*s2.V2')';

Link3 = (T\_02\*s3.V3')';

Link4 = (T\_03\*s4.V4')';

Link5 = (T\_04\*s5.V5')';

Link6 = (T\_05\*s6.V6')';

Link7 = (T\_06\*s7.V7')';

% Tool = T\_07;

% if sqrt(Tool(1,4)^2+Tool(2,4)^2)<514

% Err2 = 1;

% break

% end

%

handles = getappdata(0,'patch\_h'); %

L1 = handles(1);

L2 = handles(2);

L3 = handles(3);

L4 = handles(4);

L5 = handles(5);

L6 = handles(6);

L7 = handles(7);

Tr = handles(9);

%

set(L1,'vertices',Link1(:,1:3),'facec', [0.717,0.116,0.123]);

set(L1, 'EdgeColor','none');

set(L2,'vertices',Link2(:,1:3),'facec', [0.216,1,.583]);

set(L2, 'EdgeColor','none');

set(L3,'vertices',Link3(:,1:3),'facec', [0.306,0.733,1]);

set(L3, 'EdgeColor','none');

set(L4,'vertices',Link4(:,1:3),'facec', [1,0.542,0.493]);

set(L4, 'EdgeColor','none');

set(L5,'vertices',Link5(:,1:3),'facec', [0.216,1,.583]);

set(L5, 'EdgeColor','none');

set(L6,'vertices',Link6(:,1:3),'facec', [1,1,0.255]);

set(L6, 'EdgeColor','none');

set(L7,'vertices',Link7(:,1:3),'facec', [0.306,0.733,1]);

set(L7, 'EdgeColor','none');

% store trail in appdata

if trail == 'y'

x\_trail = getappdata(0,'xtrail');

y\_trail = getappdata(0,'ytrail');

z\_trail = getappdata(0,'ztrail');

%

xdata = [x\_trail T\_04(1,4)];

ydata = [y\_trail T\_04(2,4)];

zdata = [z\_trail T\_04(3,4)];

%

setappdata(0,'xtrail',xdata); % used for trail tracking.

setappdata(0,'ytrail',ydata); % used for trail tracking.

setappdata(0,'ztrail',zdata); % used for trail tracking.

%

set(Tr,'xdata',xdata,'ydata',ydata,'zdata',zdata);

end

drawnow

end

setappdata(0,'ThetaOld',[theta1,theta2,theta3,theta4,theta5,theta6]);

end

%%

%

%

%%

function InitHome

% Use forward kinematics to place the robot in a specified

% configuration.

% Figure setup data, create a new figure for the GUI

set(0,'Units','pixels')

dim = get(0,'ScreenSize');

fig\_1 = figure('doublebuffer','on','Position',[0,35,dim(3)-200,dim(4)-110],...

'MenuBar','none','Name',' 3D Puma Robot Graphical Demo',...

'NumberTitle','off','CloseRequestFcn',@del\_app);

hold on;

%light('Position',[-1 0 0]);

light % add a default light

daspect([1 1 1]) % Setting the aspect ratio

view(135,25)

xlabel('X'),ylabel('Y'),zlabel('Z');

title('WWU Robotics Lab PUMA 762');

axis([-1500 1500 -1500 1500 -1120 1500]);

plot3([-1500,1500],[-1500,-1500],[-1120,-1120],'k')

plot3([-1500,-1500],[-1500,1500],[-1120,-1120],'k')

plot3([-1500,-1500],[-1500,-1500],[-1120,1500],'k')

plot3([-1500,-1500],[1500,1500],[-1120,1500],'k')

plot3([-1500,1500],[-1500,-1500],[1500,1500],'k')

plot3([-1500,-1500],[-1500,1500],[1500,1500],'k')

s1 = getappdata(0,'Link1\_data');

s2 = getappdata(0,'Link2\_data');

s3 = getappdata(0,'Link3\_data');

s4 = getappdata(0,'Link4\_data');

s5 = getappdata(0,'Link5\_data');

s6 = getappdata(0,'Link6\_data');

s7 = getappdata(0,'Link7\_data');

A1 = getappdata(0,'Area\_data');

%

a2 = 650;

a3 = 0;

d3 = 190;

d4 = 600;

Px = 5000;

Py = 5000;

Pz = 5000;

%The 'home' position, for init.

t1 = 90;

t2 = -90;

t3 = -90;

t4 = 0;

t5 = 0;

t6 = 0;

% Forward Kinematics

T\_01 = tmat(0, 0, 0, t1);

T\_12 = tmat(-90, 0, 0, t2);

T\_23 = tmat(0, a2, d3, t3);

T\_34 = tmat(-90, a3, d4, t4);

T\_45 = tmat(90, 0, 0, t5);

T\_56 = tmat(-90, 0, 0, t6);

% Each link fram to base frame transformation

T\_02 = T\_01\*T\_12;

T\_03 = T\_02\*T\_23;

T\_04 = T\_03\*T\_34;

T\_05 = T\_04\*T\_45;

T\_06 = T\_05\*T\_56;

% Actual vertex data of robot links

Link1 = s1.V1;

Link2 = (T\_01\*s2.V2')';

Link3 = (T\_02\*s3.V3')';

Link4 = (T\_03\*s4.V4')';

Link5 = (T\_04\*s5.V5')';

Link6 = (T\_05\*s6.V6')';

Link7 = (T\_06\*s7.V7')';

% points are no fun to watch, make it look 3d.

L1 = patch('faces', s1.F1, 'vertices' ,Link1(:,1:3));

L2 = patch('faces', s2.F2, 'vertices' ,Link2(:,1:3));

L3 = patch('faces', s3.F3, 'vertices' ,Link3(:,1:3));

L4 = patch('faces', s4.F4, 'vertices' ,Link4(:,1:3));

L5 = patch('faces', s5.F5, 'vertices' ,Link5(:,1:3));

L6 = patch('faces', s6.F6, 'vertices' ,Link6(:,1:3));

L7 = patch('faces', s7.F7, 'vertices' ,Link7(:,1:3));

A1 = patch('faces', A1.Fa, 'vertices' ,A1.Va(:,1:3));

Tr = plot3(0,0,0,'b.'); % holder for trail paths

%

setappdata(0,'patch\_h',[L1,L2,L3,L4,L5,L6,L7,A1,Tr])

%

setappdata(0,'xtrail',0); % used for trail tracking.

setappdata(0,'ytrail',0); % used for trail tracking.

setappdata(0,'ztrail',0); % used for trail tracking.

%

set(L1, 'facec', [0.717,0.116,0.123]);

set(L1, 'EdgeColor','none');

set(L2, 'facec', [0.216,1,.583]);

set(L2, 'EdgeColor','none');

set(L3, 'facec', [0.306,0.733,1]);

set(L3, 'EdgeColor','none');

set(L4, 'facec', [1,0.542,0.493]);

set(L4, 'EdgeColor','none');

set(L5, 'facec', [0.216,1,.583]);

set(L5, 'EdgeColor','none');

set(L6, 'facec', [1,1,0.255]);

set(L6, 'EdgeColor','none');

set(L7, 'facec', [0.306,0.733,1]);

set(L7, 'EdgeColor','none');

set(A1, 'facec', [.8,.8,.8],'FaceAlpha',.25);

set(A1, 'EdgeColor','none');

%

setappdata(0,'ThetaOld',[90,-90,-90,0,0,0]);

%

end

%%

function T = tmat(alpha, a, d, theta)

% tmat(alpha, a, d, theta) (T-Matrix used in Robotics)

% The homogeneous transformation called the "T-MATRIX"

% as used in the Kinematic Equations for robotic type

% systems (or equivalent).

%

% This is equation 3.6 in Craig's "Introduction to Robotics."

% alpha, a, d, theta are the Denavit-Hartenberg parameters.

%

% (NOTE: ALL ANGLES MUST BE IN DEGREES.)

%

alpha = alpha\*pi/180; %Note: alpha is in radians.

theta = theta\*pi/180; %Note: theta is in radians.

c = cos(theta);

s = sin(theta);

ca = cos(alpha);

sa = sin(alpha);

T = [c -s 0 a; s\*ca c\*ca -sa -sa\*d; s\*sa c\*sa ca ca\*d; 0 0 0 1];

end

%%

function del\_app(varargin)

%This is the main figure window close function, to remove any

% app data that may be left due to using it for geometry.

%CloseRequestFcn

% here is the data to remove:

% Link1\_data: [1x1 struct]

% Link2\_data: [1x1 struct]

% Link3\_data: [1x1 struct]

% Link4\_data: [1x1 struct]

% Link5\_data: [1x1 struct]

% Link6\_data: [1x1 struct]

% Link7\_data: [1x1 struct]

% Area\_data: [1x1 struct]

% patch\_h: [1x9 double]

% ThetaOld: [90 -182 -90 -106 80 106]

% xtrail: 0

% ytrail: 0

% ztrail: 0

% Now remove them.

rmappdata(0,'Link1\_data');

rmappdata(0,'Link2\_data');

rmappdata(0,'Link3\_data');

rmappdata(0,'Link4\_data');

rmappdata(0,'Link5\_data');

rmappdata(0,'Link6\_data');

rmappdata(0,'Link7\_data');

rmappdata(0,'ThetaOld');

rmappdata(0,'Area\_data');

rmappdata(0,'patch\_h');

rmappdata(0,'xtrail');

rmappdata(0,'ytrail');

rmappdata(0,'ztrail');

delete(fig\_1);

end

%%

function [hout,ax\_out] = uibutton(varargin)

%uibutton: Create pushbutton with more flexible labeling than uicontrol.

% Usage:

% uibutton accepts all the same arguments as uicontrol except for the

% following property changes:

%

% Property Values

% ----------- ------------------------------------------------------

% Style 'pushbutton', 'togglebutton' or 'text', default =

% 'pushbutton'.

% String Same as for text() including cell array of strings and

% TeX or LaTeX interpretation.

% Interpreter 'tex', 'latex' or 'none', default = default for text()

%

% Syntax:

% handle = uibutton('PropertyName',PropertyValue,...)

% handle = uibutton(parent,'PropertyName',PropertyValue,...)

% [text\_obj,axes\_handle] = uibutton('Style','text',...

% 'PropertyName',PropertyValue,...)

%

% uibutton creates a temporary axes and text object containing the text to

% be displayed, captures the axes as an image, deletes the axes and then

% displays the image on the uicontrol. The handle to the uicontrol is

% returned. If you pass in a handle to an existing uicontol as the first

% argument then uibutton will use that uicontrol and not create a new one.

%

% If the Style is set to 'text' then the axes object is not deleted and the

% text object handle is returned (as well as the handle to the axes in a

% second output argument).

%

% See also UICONTROL.

% Version: 1.6, 20 April 2006

% Author: Douglas M. Schwarz

% Email: dmschwarz=ieee\*org, dmschwarz=urgrad\*rochester\*edu

% Real\_email = regexprep(Email,{'=','\*'},{'@','.'})

% Detect if first argument is a uicontrol handle.

keep\_handle = false;

if nargin > 0

h = varargin{1};

if isscalar(h) && ishandle(h) && strcmp(get(h,'Type'),'uicontrol')

keep\_handle = true;

varargin(1) = [];

end

end

% Parse arguments looking for 'Interpreter' property. If found, note its

% value and then remove it from where it was found.

interp\_value = get(0,'DefaultTextInterpreter');

arg = 1;

remove = [];

while arg <= length(varargin)

v = varargin{arg};

if isstruct(v)

fn = fieldnames(v);

for i = 1:length(fn)

if strncmpi(fn{i},'interpreter',length(fn{i}))

interp\_value = v.(fn{i});

v = rmfield(v,fn{i});

end

end

varargin{arg} = v;

arg = arg + 1;

elseif ischar(v)

if strncmpi(v,'interpreter',length(v))

interp\_value = varargin{arg+1};

remove = [remove,arg,arg+1];

end

arg = arg + 2;

elseif arg == 1 && isscalar(v) && ishandle(v) && ...

any(strcmp(get(h,'Type'),{'figure','uipanel'}))

arg = arg + 1;

else

error('Invalid property or uicontrol parent.')

end

end

varargin(remove) = [];

% Create uicontrol, get its properties then hide it.

if keep\_handle

set(h,varargin{:})

else

h = uicontrol(varargin{:});

end

s = get(h);

if ~any(strcmp(s.Style,{'pushbutton','togglebutton','text'}))

delete(h)

error('''Style'' must be pushbutton, togglebutton or text.')

end

set(h,'Visible','off')

% Create axes.

parent = get(h,'Parent');

ax = axes('Parent',parent,...

'Units',s.Units,...

'Position',s.Position,...

'XTick',[],'YTick',[],...

'XColor',s.BackgroundColor,...

'YColor',s.BackgroundColor,...

'Box','on',...

'Color',s.BackgroundColor);

% Adjust size of axes for best appearance.

set(ax,'Units','pixels')

pos = round(get(ax,'Position'));

if strcmp(s.Style,'text')

set(ax,'Position',pos + [0 1 -1 -1])

else

set(ax,'Position',pos + [4 4 -8 -8])

end

switch s.HorizontalAlignment

case 'left'

x = 0.0;

case 'center'

x = 0.5;

case 'right'

x = 1;

end

% Create text object.

text\_obj = text('Parent',ax,...

'Position',[x,0.5],...

'String',s.String,...

'Interpreter',interp\_value,...

'HorizontalAlignment',s.HorizontalAlignment,...

'VerticalAlignment','middle',...

'FontName',s.FontName,...

'FontSize',s.FontSize,...

'FontAngle',s.FontAngle,...

'FontWeight',s.FontWeight,...

'Color',s.ForegroundColor);

% If we are creating something that looks like a text uicontrol then we're

% all done and we return the text object and axes handles rather than a

% uicontrol handle.

if strcmp(s.Style,'text')

delete(h)

if nargout

hout = text\_obj;

ax\_out = ax;

end

return

end

% Capture image of axes and then delete the axes.

frame = getframe(ax);

delete(ax)

% Build RGB image, set background pixels to NaN and put it in 'CData' for

% the uicontrol.

if isempty(frame.colormap)

rgb = frame.cdata;

else

rgb = reshape(frame.colormap(frame.cdata,:),[pos([4,3]),3]);

end

size\_rgb = size(rgb);

rgb = double(rgb)/255;

back = repmat(permute(s.BackgroundColor,[1 3 2]),size\_rgb(1:2));

isback = all(rgb == back,3);

rgb(repmat(isback,[1 1 3])) = NaN;

set(h,'CData',rgb,'String','','Visible',s.Visible)

% Assign output argument if necessary.

if nargout

hout = h;

end

%%

end

end

% Finally.

**Kaynaklar**

1. <https://www.mathworks.com/matlabcentral/fileexchange/14932-3d-puma-robot-demo>
2. <https://www.sciencedirect.com/science/article/pii/S1474667016373207>
3. https://www.youtube.com/watch?v=mN8KuZRYhbc